

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant:	Scott James Weaver	§	Group Art Unit:	2193
		§		
Serial No.:	10/041,743	§		
		§	Examiner:	Tuan A. Vu
Filed:	01/10/2002	§		
		§		
For:	DATA WEDGE	§	Atty. Dkt. No.:	9288

Mail Stop: AF
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

PRE-APPEAL BRIEF AND REQUEST FOR REVIEW

Dear Sir:

Applicant requests a pre-appeal review of the final rejection, in this case. No amendments are being filed with this request. This request is being filed with a Notice of Appeal.

102(e) Rejection of All Claims (17-33) by Tamboli (US Pat. No. 6,792,431)

Tamboli teaches a data integration method using a Dynamic Common Model where “customers typically need to locate data in a source repository, transform the data from a source format to a destination format, and transfer the data from the source to the destination.” (Col. 1, lines 22-25.) Tamboli specifically fails to show or suggest “creating a first schema comprising the data model of the first software component” and “integrating the first schema into a data wedge,” as required by Applicant. The Office appears to assert that the “data wedge” is equivalent to the Dynamic Common Model of Tamboli on page 2, section 3, paragraph 3 by stating “integrating the first schema into a data wedge (e.g. Dynamic Common Model 118-Fig 2).” Applicant disagrees with this assertion. Applicant teaches that a “data wedge” is a software component (see

specification, paragraph 21.) and further teaches that a software component has “source code” (see specification paragraph 2). A person of ordinary skill in the art would clearly recognize Applicant’s data wedge as a type of computer program having software codes that are executable by the computer. Conversely, Tamboli teaches that the dynamic common model is “an aggregate of all mappings to and from native formats and dynamic common formats within a data integration application.” (Col. 7, lines 36-38.) The dynamic common model is a collection of mappings for data represented in different formats. There is no teaching that the dynamic common model contains any software or executable code or anything other than data (the mappings). Therefore, it cannot be a software component or a “data wedge,” as required by Applicant.

Furthermore, Tamboli provides support for this position when he describes one benefit of using a dynamic common model. Tamboli describes a case where “a user redefines an implementation of a datatype” causing a failure to fully integrate the data. Tamboli teaches a two-step repair process where the second step involves updating the data mappings in the dynamic common model. Tamboli teaches “to the extent that the mapping needs to be amended, no programming is required, only text editing.” (Col. 25, lines 29-30.) Tamboli does teach that code in some applications may need to be changed but specifically points out that **no programming** is required to change the dynamic common model. It is clear from the teachings of Tamboli that the dynamic common model does not contain computer or programming code but instead consist of text used to map data. The Office has improperly equated Applicant’s “data wedge” with Tamboli’s dynamic common model. Thus, at least this element is missing from Tamboli. The Office’s rejection is therefore improper and Applicant ask that it be withdrawn.

Tamboli also fails to show or suggest a “first software component” or “the data model of the first software component,” as required by Applicant. The Office asserts that the “First Native” shown in FIG. 1 is equivalent to Applicant’s first software component by stating “of the first software component -- see First Native 106 – Fig 1.” (Page 2, section 3, third paragraph.) Applicant disagrees. The full name for element 106 in FIG. 1 is “First Native Repository.” In Col. 6, lines 44-66, Tamboli defines the term “repository” as data records or data stores and teaches that “‘Native repositories’ are data stores outside a data integration application” As shown above, a software component

has programming code and is considered software. Tamboli specifically defines “native repositories” as data stores. A person of ordinary skill in the art would conclude that native repositories are just structures containing data and not software or a software component as required by Applicant. The Office has therefore failed to show a first software component and in doing so has also failed to show a data model of the first software component. At least these elements are missing from Tamboli, which renders the rejection improper.

Tamboli fails to show or suggest “creating a second schema comprising the data model of the second software component,” as required by Applicant. In regard to these elements, the Office states “e.g. Fig. 1: adapter 124, Transform 206, mappings 120 – Note: mappings and adapter transformation for the second native format reads on XML schema model for second software component.” (Page 2, section 3, forth paragraph – page 3, first paragraph.) Applicant is at a loss to understand how this statement specifically points out the elements of Applicant’s claim. Applicant first requires a second software component. The Office appears to suggest that the “second native format” is equivalent to Applicant’s second software component. As shown above, a native format is not a software component. The Office has failed to show this element in Tamboli’s teachings. Next, there must be a data model of the second software component. This is also missing. Finally, the Office has failed to show a schema comprising the data model of the second software component.

The mappings and adapter transformations do not equate to any of these elements. Tamboli teaches that “‘Adapters’ are implementations of interfaces between native repositories and other elements of embodiments...” Applicant clearly requires a “data model of a software component.” Adapters are just implementations of interfaces. Furthermore, the mappings are only a snap shot of the data repositories. There is no teaching by Tamboli that a data repository is a data model of an adapter. In fact, as shown above, Tamboli teaches that data in a repository can be redefined without making an update the data mappings and adapter. Tamboli only teaches that the redefined data causes a problem when an adapter transfers data to another repository. There is no teaching that redefining data causes any problems in the repository or with software that controls the repository. This implies that the data model is not part of the adapter since

the data model can change and continue to function outside the adapter. This would not be possible if the data model was part of the software component (adapter). A person of ordinary skill in the art would have to conclude that mappings and an adapter are not the same elements required by Applicant. Therefore, at least these elements are missing from Tamboli and the rejection is improper.

102(e) Rejection of All Claims (17-33) by Worden (US Pat. Pub. 2003/0149934)

With respect to independent claims 17 and 25, the Office failed to address “integrating the second schema into the data wedge,” as required by Applicant. With respect to independent claims 17, 25, and 33, the Office failed to address “transferring the translated data element to the second software component when a function of the second software component is called by the first software component,” as required by Applicant. Applicant requires these elements but the Office has failed to provide any evidence that shows or suggests that Worden teaches them. The Office has failed to establish a *prima facie* case of anticipation. Therefore, the rejection is improper.

Worden fails to show or suggest “creating a first schema comprising the data model of the first software component,” as required by Applicant. In regard to these requirements, the Office states “e.g. Fig. 10: XML(1), Schema (1); para 0014-0119.” (Office action, page 8, second paragraph.) Again, Applicant is at a loss to understand the Office’s rejection. It is not clear what element, in paragraphs 14-119 of Worden, the Office has equated to a first software component. The cited paragraphs start on page 2 part way through the DESCRIPTION OF THE PRIOR ART section, continues through and includes all of the SUMMARY OF THE PRESENT INVENTION and BRIEF DESCRIPTION OF THE FIGURES sections and ends part way into the DETAILED DESCRIPTION section in the middle of several examples. Paragraph 14 teaches “XSL is a complex Programming Language.” Paragraph 15 discusses a “significant problem of version control between changing XML-based languages.” The Office also cited Fig. 10 but the cited paragraphs do not even contain Worden’s detailed description of Fig. 10. Applicant fails to understand the relevance of the cited material and the Office has failed

to provide details sufficient for a reasoned response. The Office has thus failed to establish a *prima facie* case of anticipation. Therefore, the rejection is improper.

Note: The space restrictions of the pre-appeal brief format do not permit Applicant to address all of the defects of the current Office Action.

CONCLUSION

For one or more of the above reasons, the Office has failed to properly reject the claims. Applicant asks that the Office reconsider this application and allow all pending claims. Please charge any fees that might be due, excluding the issue fee, to deposit account 14-0225.

Respectfully submitted,

Date: April 4, 2007
(Electronically Filed)

/Harden E. Stevens, III/
Harden E. Stevens, III
Reg. No. 55,649

NCR Corporation
1700 South Patterson Blvd.
Dayton, Ohio 45479

(803) 939-6505
(803) 939-5521 (fax)
ss133111@ncr.com (email)